WEB-BASED SECURITY AND FILTERING SYSTEM WITH PROXY CHAINING

This invention relates to controlling web-based network communications on the Internet and more particularly it relates to a customizable system for both filtering web-based Internet requests for outbound access to web sites and for controlling web-based Internet requests for inbound access by others of a web site of the system.

The World Wide Web provides minimal security protection for its users and service providers. Major service providers create expensive and complex security zones to control the inbound user access. The cost prohibitive and complex nature of this solution is not suitable for a growing number of small business with simple computer systems.

Different security threats exist for Internet users where WWW browsers can provide unlimited access to many unsuitable resources, unsuitable for users within a certain age and preference. Occasionally, the highjacking of the user browser's control poses annoying problems. Parents having Internet enabled computers at home feel helpless and concerned about the future of their children who may access inappropriate adult Internet content in their absence.

The present invention addresses these concerns and more and it provides, in summary a combination customizable system for filtering web-based requests for outbound access to web sites and for controlling web-based requests for inbound access by others of a web site of the system. It includes an administrative module that includes configuration settings for inbound and outbound communications, list maintenance functions and that is located in a user computer for configuring a range of access levels from maximum access to full suspension and having administrator account, regular accounts and one anonymous guest account to be used by general users with no system-based account. If a user attempts to access an unfriendly site the user is

-1-

anonymously forwarded to a friendly site. A first proxy server which is a firewall system in each

user computer of a local area network has access to the world wide web, has a customizable

friendly and unfriendly outbound list, one of which is active and a customizable friendly and

unfriendly inbound list, one of which is active. A second HTTP proxy server exists between the

first proxy and the Internet located as a gateway of a Local Area Network system or at an

Internet Service Provider, the second proxy server being capable of communicating to a HTTP

proxy server of a destination or directly to a destination and having an Internet Protocol address

known only to the system's first proxy server administrator. The system automatically generates

a logfile of all web sites visited by each user. For single users in a home or office the second

proxy is provided through the Internet Server Provider.

OBJECTS AND ADVANTAGES

The following are important objects and advantages of the present invention:

(1) to provide a filtering and monitoring system for HTTP network communications,

(2) to provide such a system that is for both outbound and inbound HTTP

communications,

(3) to provide such a system that is tamper-proof,

(4) to provide such a system that uses an administrative module that has unfriendly and

friendly lists for both inbound and outbound communications,

(5) to provide such a system wherein the administrative module is an interface located in a

user computer for configuring a range of access levels applicable to the user and wherein the

range of access levels ranges from maximum unlimited access to full suspension and the

administrative module is capable of creating three types of user accounts that have unique user

-2-

3

names and passwords for each account including (a) an administrator account that is self-configuring, meaning having the full power to do the configuring (b) regular accounts with administrative privileges other than the privilege or power to create additional accounts or view information on any other accounts and (c) regular accounts without administrative privileges and in addition a fourth type of user account namely one anonymous guest user account to be used by all general users having no system-based user name or password (the word "system" meaning the system of the present invention),

(6) to provide such a system wherein the administrative module includes configuration settings for inbound communications and for outbound communications and has list maintenance functions that including list editing, list deleting, searching of lists, saving of lists, proxy chaining routing, adding and deleting users, interchanging lists and importing and exporting lists,

(7) to provide such a system that is suitable for persons of all ages,

(8) to provide a system of filtering and monitoring both inbound and outbound web-based Internet communications that employs proxy chaining,

(9) to provide such a system wherein the inbound communications is set up so that the actual location of the highly sensitive resources are located in an unpublished location that is the replacement location to which are rerouted requests rejected by the system's first proxy server and wherein approved users' hosts are listed in the first proxy server in the unfriendly inbound list are bounced by the first proxy server to the alternative location at the unpublished address having the resources and users' hosts who need to be rejected for access are assumed by the proxy server to be friendly and have their request sent to a published address that contains harmless information,

-3-

(10) to provide a system of filtering and monitoring that is suitable for a single user or for groups of users and that is suitable for HTTP over TCP/IP communications protocols and

(11) to provide such a system suitable for users having a dialup modem connection to the Internet and users having Internet access through network lines (direct Internet access).

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A shows the Outbound client's HTTP requests and system of the present invention.

FIG. 1B shows the bypassing of the system.

FIG. 2 shows the interception of inbound requests to protect internal resources.

FIG. 3A shows interception of outbound HTTP Requests without proxy chaining with network line access.

FIG. 3B shows the same as FIG. 3A except for dialup modem connections.

FIG. 4A shows the use of proxy chaining for outbound interception enforcement for single users with network line access

FIG. 4B shows the use of proxy chaining for outbound interception enforcement for single users with dialup modem connections.

FIG. 5A shows the use of proxy chaining with one deployed Gargoyle or first proxy server for outbound interception enforcement for a group of HTTP clients with every web user using the first proxy server's account and having the same level of access and configurations.

FIG. 5B shows the use of proxy chaining for outbound interception enforcement for a group of HTTP clients with network line access with each computer having its own instance of the system of the present invention deployed and has its own user account and where every user account may have varying levels of access.

-4-

5

FIG. 6 shows a diagram of a sequence of events when outbound communications depicted as in FIGS. 1A and 4A are intercepted and then approved and forwarded.

FIG. 7 is a diagram of a sequence of events when outbound communications depicted as in FIGS. 1A and 4A are intercepted and denied and either dropped or forwarded to a replacement URL.

DETAILED DESCRIPTION OF THE DRAWINGS

The system of the present invention addresses both inbound and outbound web-based communication requests. Before describing the Figures herein, the following terms are clarified. HTTP means Hypertext Transfer Protocol which is the protocol used for web-based interprocess network communication.

All the HTTP applications are software processes that are HTTP "protocol" for interprocess network communications. HTTP clients are HTTP applications that can receive user or process generated inputs and generate outputs and may provoke actions or events on other software applications such as web servers. The inputs are either generated through use of Graphical User Interface (GUI) tools or via other software processes, or are self-generated by the client application. Actions are mostly in response to client requests for statically or dynamically created formatted files.

The term "HTTP Browser" as used in this application refers to commercially available software applications of a user that handles the user's requests to go onto the Internet. Examples of HTTP Browser software include those sold under the name Netscape Navigator and Microsoft Explorer. HTTP Browsers are located in and may be launched from the user's computer after being installed there. In the description of the system of the present invention, we sometimes call

-5-

the HTTP browser the "HTTP Client" because it is the browser, not the user, that deals with the web server and there are many browser's making requests to a particular web server. It is important to note, however, that the browser applications are only one type of HTTP client. The system of the present invention is suitable for all types of HTTP clients and is not limited to browser-based deployment.

A "web server" is the software application that manages someone's web site. The web server serve many "clients" by responding to requests from the "clients" for resources (the web site) and forwards the resources to the clients or refers the request to another web server.

The term "router" means the connectors that are used to link different networks together. Routers can direct or route information to the correct destination in cyberspace.

The term "gateway" is often used to link two different networks types together, for example the transfer of information between a Macintosh network and a PC network. A gateway is usually a computer that acts as a connection between a private internal network another network such as the Internet. Gateways may be used as firewalls. When gateways are used as firewalls they can transmit information from an internal network to the Internet. Gateways can also examine incoming information to determine if the information should be allowed entry to the network.

The term "proxy server" represents a destination to the client and it is a piece of software application. All proxy server's referred to in this patent application are HTTP-based proxy servers which use HTTP protocol for interprocess network communication.

The first proxy server of the system of the present invention is sometimes called by its trade name "Gargoyle". A user would configure his or her browser to use our system's first

-6-

7

proxy server (Gargoyle) and place it intermediate between the user's browser and the network (the Internet). This is done by inserting the correct settings in the browser settings window such as the host's IP address and port number that the proxy server is accepting client requests from (running on). The host can be although it does not necessarily have to be the local host, i.e. the same computer that the browser is deployed on.

In computer science terms, the Gargoyle dispatcher or scheduler runs with the "main process thread". The dispatcher starts a server socket and waits for client socket connection requests. Upon receiving such requests, the server socket opens a client socket for that particular connection and spawns a worker thread and passes that socket handler to that worker thread. After the HTTP client (browser) receives the requested resources, the dispatcher terminates the working thread. Accordingly, only the main thread continues running indefinitely until the user terminates the application process and closes out of Gargoyle (the first proxy server).

In the Figures of the present invention "WS" refers to a work station; "PC" refers to a personal computer, "LAN" refers to a local area network, "PPP" is defined as point to point protocol which is the kind of protocol used for modem-based communication; and "TCP" means transmission control protocol.

A friendly list means a list of preferred names of entities such as URLs or subsets of URLs. A match with a friendly list results in the requested URL being forwarded. An unfriendly list is a list of non-preferred names of entities such as URLs or subsets of URLs. A match with an unfriendly list results in the request not being forwarded to the destination - instead being terminated or forwarded elsewhere.

FIG. 1A shows the Outbound client's HTTP requests in the system of the present

invention. The large box on the left of FIG. 1A is represented in subsequent Figures as a picture of a work station or computer. As seen from FIG. 1A, the following steps occur when the local user wishes to visit a web site on the World Wide Web. First,.the HTTP browser/client forwards the request for a URL, which is called herein an "outbound communication" to the system's first proxy server (Gargoyle). Second, Gargoyle is programmed to automatically register the request in a logfile of all web sites requested by that user account.

Third, Gargoyle checks the identity of the requesting client. If it is an internal client then the outbound list is used as a reference point and if the client is an external client then the inbound list is used as a reference point. Gargoyle then checks the identity of the requested URL against the friendly or unfriendly list as the reference point and which is being maintained by Gargoyle for that particular user account based on the configured access rules. The list used as a reference point is outbound if the request is an outbound request and the client is internal and the list used as a reference point is inbound if the request is inbound and the client is external.

The fourth step is to process the domain authentication based on the configured access security rules. For example, if there is found a partially or fully successful match between the requested URL and any list items, then if the list is unfriendly the request is not acceptable and if the list is friendly, the request is forwarded appropriately to the next proxy server or destination web server. On the other hand, if no match is found then if the list is friendly the request will be denied and if the list is unfriendly the request will be accepted and processed appropriately. This step is called domain authentication within the first proxy server's (Gargoyle's) main processing thread. By a "partially successful match" is meant the following: let us say that the letter string "SEX" is in the unfriendly outbound list, if one of the requested URL has those letters in even part

-8-

of the URL, it will trigger a successful match outcome.

If the request is approved based on these rules the request is forwarded and if rejected the request is either terminated or re-routed without the knowledge of the user to an alternative or replacement destination link. Where the request is not blocked, the fifth step is that the request goes to the Internet in one of two ways. The first way is direct Internet and it is described in detail in FIG. 3A - the request goes directly to a LAN or a network line. The second way is by dialup modem and is depicted in FIG. 3B.

FIG. 1B is generally similar to FIG. 1A except that in FIG. 1B the hypothetical user is seen to bypass the system of the present invention by virtue of the fact that the user reconfigured the browser so that it does not make use of the first proxy server. Note that in FIG. 1B Gargoyle is not in the path to the Internet. This bypass problem is relevant for any user that is not a child so young that it cannot do this. Furthermore, the system of the present invention solves this problem by means of the use of double proxy servers, as described in further detail in FIGS. 4A, 4B and 5A, 5B.. As seen in FIG. 1A, this variation is suitable for children who could not reconfigure the web browsers to use direct network access without the first proxy server, which is a firewall system proxy server. As seen in FIG. 1B, existing browsers such as Netscape Navigator or Microsoft Explorer are simply configured to use the direct Internet access.

FIG. 2 shows the system's interception of Inbound HTTP Requests to protect internal resources of the system. The address of servers having highly sensitive resources is unpublished and requests are sent to a server with a published address having no significant resources. Before that happens, however, the request is first sent by the system to the first proxy server to be checked against the friendly or unfriendly list. If the proxy server simply checked the requests for

approval and allowed through those requests that are from paid users who are supposed to get

approved for access to the resources, then the resource web site's address would have to be

published. The result would be that highly sensitive resources would be vulnerable to access by

intruders who "hack" into the web site and obtain the sensitive information.

Instead, as seen in FIG. 2 in accordance with the system of the present invention, the

inbound filtering system operates in a reverse manner. The actual location of the highly sensitive

resources are located in an unpublished location. That location is the replacement or alternative

location to which are rerouted requests rejected by the system's first proxy server. Therefore,

users that paid for access to the premium highly sensitive documents ("approved users") and who

should get approved for access are listed by their trusted host name in the proxy server's

"unfavorite" or "unfriendly" inbound list. As a result, the proxy server bounces the user' URL

request to the alternative/replacement default location at the unpublished address having the

resources. Conversely, users who did not pay for premium access ("unapproved users") and who

need to be rejected for highly sensitive access are assumed by the proxy server to be "friendly" or

"favorite" and have their URL request approved. But that approval means that their request is

sent to the host that has the published address and that host does not in fact contain the sensitive

information but instead contains harmless information. Consequently, inbound access to sensitive

information is monitored and filtered.

FIG. 3A and 3B describe the monitoring of outbound communications without the use of

proxy chaining to ensure that the users do not bypass the system's first proxy server. This

solution is appropriate for very small children who cannot reconfigure their browser to bypass

Gargoyle. FIG. 3A is for direct Internet access and FIG. 3B is for dialup modem access to the

Internet. This system, which is without the additional proxy server, can be fairly effective only for children within certain ages in filtering outbound requests because few children know how to disable the browser's configuration to connect to the Internet directly and few know how to configure their browser's proxy configuration. In FIG. 3A, the Gargoyle proxy server is installed on every computer in the LAN1. Where a request is blocked it may be rerouted to an alternative web site or simply terminated. If re-routed the user will be unaware that it was rerouted.

FIG. 4A shows the use of proxy chaining to improve security further for outbound communications by a single user using interception enforcement. In order to create the most tamper-proof web-based Internet access monitoring system for general users, the systems' first proxy server can provide the complete personalized filtering system for the entire family. The solution is based on using two or more interconnected HTTP proxy servers. The first proxy server is the firewall system (Gargoyle) and is configured to be connected to another HTTP proxy server. The second proxy server can also be a firewall system proxy server but with empty unfriendly outbound access list ("unfriendly outbound list"). This is called proxy chaining. The key to the successful operation is that the Internet Protocol address (meaning, the identity) of the computer that hosts the second proxy server is unknown to general users and is disclosed and known only to the administrator account user who configures the first proxy server. Only the second HTTP proxy server and no other proxy server is configured to process HTTP communications to the LAN gateway (or router) and thus to the Internet because the second HTTP proxy server is only proxy server that knows the address and location of the LAN gateway (or router).

The first proxy is located either on the LAN (for example LAN1) with respect to the user

computer (if one exists) or on the same computer host that the browser is installed and the user is using. The location of the second HTTP proxy server can vary. The second HTTP proxy server may be located on either of the ISP-based LANs, called LAN1 or LAN2. Most commonly, the user computer is connected to the ISP through a modem connection and the first proxy server is installed and configured on this single computer.

The system provides personalized outbound filtering for numerous user's accounts on stand alone or networks of personal computer work stations. Select an ISP who can provide an HTTP proxy based Internet connection either through modem or network line access. Under this system, the only way to connect to the Internet is to first connect to the first proxy server (Gargoyle) and then be forwarded to the ISP-based HTTP proxy server, then to the Internet. The ISP-based proxy Internet Address and port number are provided to the customers (who will be the system administrator) and are not available to the general public.

The HTTP client (the browser) should be configured to use the system's first proxy server (Gargoyle) as its HTTP proxy server located on either the local host or anywhere on the LAN and the first proxy server should be further configured to forward requests to the second proxy server. To do this using the administrative tools, the user just selects to forward the outbound requests to a different proxy server (use the ISP proxy server as the second proxy server) The administrative user creates one anonymous user account with the highest restriction and no administrative roles which is to be used by all users with no valid system accounts and especially young children or general users in public locations, creates a few user accounts with less restricted accounts and no administrative roles for older age children and creates a few user accounts with administrative rights most suitable for parents. Only the user accounts with administrative rights can inspect and

see the ISP related configurations. The other users may not reconfigure their HTTP clients (browsers) to directly use the ISP provided proxy server. Consequently, these users have no way of bypassing the first proxy server and get Internet access without any filtering mechanism. The system administrative tools will hide the ISP proxy location and thereby thwart any users who try to bypass the first proxy server. Furthermore, the location of the LAN gateway (or router) is unknown to all web users and all the proxy servers except the second proxy server.

The system's first proxy server delegates client's HTTP requests to other proxy servers. To avoid extra costs of having your own second proxy server, the second proxy server is located in the Internet Service Provider ("ISP"). Now since the address of that second proxy server is listed in Gargoyle but not otherwise known to the user, if the user bypasses the first proxy server that user will still not know the address of the ISP proxy server and consequently still will not be able to get network access through the ISP or to the ultimate URL requested. While FIG. 4A is for direct Internet access, FIG. 4B is the same but it is for the situation where there is dialup modem access to the Internet.

FIG. 5B illustrates use of the system in the same manner as FIG. 4A except that FIG. 5B deals with a solution for a group of HTTP clients users rather than for a single user. An example of a group would be a school. In this case, each user computer has a first proxy server (Gargoyle). The second proxy server does not need to be Gargoyle - in other words it does not need to have the friendly and unfriendly list information. It does not need to because the system assumes that any effort by the users to bypass the first proxy server located in the user computer requires the request to be rejected. The second proxy server runs on the host computer system with a secret address unknown to the users and it is located in the LAN in the path to the Internet.

-13-

The second proxy server may be a firewall proxy of the LAN although not necessarily the first proxy server of the system of the present invention.

FIG. 5B deals with the situation where there are many users in a public computer site such as a public school. As seen in FIG. 5B, the complete personalized filtering system of the present invention provides a similar tamper-proof web-based Internet access monitoring system for public computer sites running many computers on a local LAN. In this case, a local user may request for a URL, and if the URL passes the anonymous user outbound access list (whether friendly or unfriendly), it will be routed to the second proxy server on the LAN before reaching the Internet. The first proxy server, the firewall system proxy server, routes the request only to the known host (as defined by its configuration). This is the only way of accessing the Internet. The first proxy server is installed on every computer of the LAN1. All of them are configured to use proxy forwarding (proxy chaining) at the address and port on the "Host "A"" acting as a gateway.

On occasion the user computers upon connection to the network may receive a dynamic IP address. This dynamically allocated IP address does not affect the internal client since the browser is configured to use the first proxy server at the IP address 127.0.0.1 rather than a particular IP address. Furthermore, the first proxy server will also identify the internal client by comparing incoming client IP address and local host address. If those two addresses are the same, the client is an internal client and the outbound list will be used for security interception.

All the browsers are also configured to access the first proxy server by the unique static Internet Protocol ("IP") address of their computer or by using local host IP address127.0.0.1 for dynamically allocated IP addresses. Also the browsers are configured to access the selected port number on which the first proxy server's main thread is tending to the client requests. Any other

-14-

attempt to reach the Internet will fail since the browser users are not aware of the address of "Host "A"" and therefore they cannot reach the Internet directly.

The second proxy server is configured to forward the incoming requests to the router or gateway, and further to the Internet. In some LAN setups although not the one shown in FIG. 5B, the host where the second proxy server is running may have two network cards configured with two distinct IP addresses for network connectivity, one being connected to the LAN 1, and the other to the Internet.

FIG. 5A also deals with a group of HTTP clients but is different from FIG. 5B. Unlike FIG. 5B, FIG. 5A has the Gargoyle proxy server not in each student's workstation computer but rather in the path to the Internet. Consequently, every user (student) has the same list and the same level of access provided through one single user account under which Gargoyle will be launched.

FIG. 6 (which refers to outbound communications as depicted in FIGS. 1A and 4A) shows a diagram of a sequence of technical events that occur whenever outbound communications of one request by any system are intercepted and then approved and forwarded.. In both FIG. 6 and FIG. 7, time runs forward from the top of the Figure to the bottom of the Figure and each of the arrows represents an event or step that occurs while using the system of the present invention. The beginning of each arrow indicates who or what initiates the event.

Each step of FIG. 6 is represented by a numeral. Numeral 1 shows that the user launches the first proxy server (Gargoyle) at a known port and host. Numeral 2 shows the first proxy server thread waiting for HTTP client socket connections. Numeral 3 shows the user entering the URL on his web browser and sending the request by clicking on the "Enter" key. Numeral 4

-15-

shows the first proxy server's socket accepting the HTTP client (browser)connection request where the HTTP client opens a client socket and forwards its request to the first proxy server. Numeral 5 shows the first proxy server registering the request in a logfile (one logfile per user account). Steps 6, 7 and 8 are the domain authentication. Between numeral 4 through numeral 6 the first proxy server uses the incoming client address to identify if this is an outbound request or an inbound request. If the requesting client address is the one which is uniquely configured to be the protected host IP address in the user account or it is the same as the local host IP address, the first proxy server will identify that client as an internal client, and the request is identified as outbound request. Otherwise, it will use the inbound access list. If a communication originates from an inbound client it is called an "outbound communication" and uses the outbound access list; if not it is called an "inbound communication" and uses the inbound access list.

In FIG. 6, numeral 7 shows the first proxy server uses the outbound access list to find a full or partial match between the client's requested destination URL and the list items. After Numeral 8, the results of comparisons will be used by the first proxy server to decide either to forward or stop the request. Numeral 9 shows the first proxy server will establish a client socket to the destination, or possibly will forward the connection request to the next proxy server in the proxy chaining. Numeral 10 shows the connection request will be forwarded to the destination server. Numeral 11 shows the destination host will grant the connection request and send back the socket handler. Numeral 12 shows the proxy server will forward the socket handler to the first proxy server. Numeral 13 shows the server thread spawns a working thread sub-process and passes these two socket handlers from the client, and to the destination or next proxy server to that thread. Numeral 14 shows that the worker thread will use these two socket handlers to

-16-

17

mediate the communication. Numerals 15 and 16 show the HTTP client request has been

forwarded to the destination web server. Numeral 17 shows the destination web server will search

and allocate the requested resources. Numeral 18 shows the web server will return the resources

back to the intermediate proxy servers. Numeral 19 shows the resources are forwarded to the first

proxy server. HTTP client will receive the resources as seen by numeral 20. The client closes the

communication channel by closing the socket to the system's first proxy server (Gargoyle) as seen

by numeral 21. Numeral 22 shows Gargoyle in turn will close the communication channel.

Numeral 23 shows all the remaining proxy servers wil close their socket connections. Gargoyle

stops the working thread as seen by numeral 24. Numeral 25 is the same as numeral 2 - the first

proxy server thread is waiting for more HTTP client socket connections.

FIG. 7 (which refers to outbound communications that are depicted in FIGS. 1A and 4A)

is a diagram of a sequence of events when outbound communications are intercepted and then

denied whereupon they are either dropped or forwarded to a predefined friendly replacement

URL. In FIG. 7, numerals 1 through 8 (and 4A through 8A) are the same as in FIG. 6 when the

outbound communications are approved. Numeral 9 shows the case where Gargoyle uses the

security rules (comparing the requested URL against outbound friendly or outbound unfriendly

lists) and as a result the HTTP client request is denied and terminated. Numeral 10 shows an

error message is created and sent to the HTTP client and that the HTTP client socket is closed by

the first proxy server. Numeral 11 shows the first proxy server's main thread continues waiting

and listening to incoming requests via the server socket. Numeral 9A shows the first proxy server

attempts to connect to the friendly alternative destination host and tries to open a client socket to

that server. Numeral 10A shows the new destination host will accept the request and will open a

-17-

socket for this connection and returns the socket handler. Numeral 11A shows the first proxy server spawns a working thread (sub-process running as a thread) per client request. Numeral 12A shows the server passes these two sockets to the worker (or working) thread, and the worker thread will connect these two sockets from HTTP client to the destination host and web server. The working thread will channel packets from destination to the client. Numeral 13A shows the web server will obtain the requested resources. Numeral 14A shows the destination host will forward these resources to the client through the opened communication path. Numeral 15A shows that after receiving the resources, the client closes the socket. Client will close the connection after facing any networking errors or if the server closes the connection. Numeral 16A shows the proxy server closes the socket to the destination. Numeral 17A shows the main scheduler thread of the first proxy server will stop this thread.

As seen in FIG. 7, denoted by numerals 2, 11 and 18A, the server will continue waiting to accept further requests and will spawn working threads concurrently as many as needed to serve the client requests. Therefore, after the end of any thread life cycle, the main server socket in the first proxy server is open for further client requests.

It is contemplated by the proxy chaining described herein that there can be a third proxy server or even additional proxy servers that forward inbound requests for resources to other proxy servers.

The system of the present invention operates for HTTP communications and is compatible with both telephone lines and direct network lines. The system of the present invention allows use of a friendly list and an unfriendly list for both inbound and outbound communication filtering. The system employs an administrative module which is an interface located in a user computer for

-18-

configuring a range of access levels applicable to the user. The range of access levels ranges from

maximum access, which is 100% full access, to full suspension. The administrative module is

capable of creating three types of user accounts that have unique user names and passwords for

each account, including (a) an administrator account that is self-configuring, (b) regular accounts

with administrative privileges other than the privilege or power to create additional accounts or

view information on any other accounts and (c) regular accounts without administrative privileges

and in addition a fourth type of user account which is an anonymous guest user account to be

used by all general users having no system-based user name or password. The term "self-

configuring" used herein simply means having the power to do the configuring on his own.

Every instance of launching the first proxy server must be done under a user account

recognized by the system of the present invention or under an anonymous guest account. More

than one first proxy server can be launched at the same time on the same host computer but one

would have to use separate port numbers.

The administrative module includes configuration settings for inbound communications

and for outbound communications and has list maintenance functions that including list editing,

list deleting, searching of lists, saving of lists, proxy chaining routing, adding and deleting users,

interchanging lists and importing lists. Importing of lists means that a user can easily transfer a list

of web sites from a flat file into the system's administrative module. Exporting of lists means that

after you create the list you can save it in a flat file on the system and you may send it to other

users to be used as their lists.

The system of the present invention provides that each user account has the following

attributes that can be configured. Any user having administrative privileges can populate an

inbound list, can populate an outbound list, determine and select whether a list will be either a friendly list or an unfriendly list. That user can also, if using proxy chaining, define the host address where the next forwarding proxy server is located and that proxy server's port number. For the protection of additional HTTP clients that are not internal clients, one can also select a unique domain name or host IP address to be protected by the same first proxy server and thus treat all HTTP clients from that domain name or from that host address as an internal client. The first proxy server will use this protected address and local host IP address to identify the internal HTTP client and to properly use inbound and outbound lists. One can also select a port number which the first proxy server's main server socket will use to accept the incoming HTTP client request. This is the same port number which was used in the user's browser's proxy configuration.

All information contained or used within the administrative tools or used by the first proxy server are encrypted for additional security measure.

Although the invention has been described in detail in the foregoing specification and accompanying drawings with respect to various embodiments thereof, these are intended to be illustrative only and not limiting. One skilled in the art will recognize that various modifications and variations may be made therein which are within the spirit and principles of the invention and the scope of the appended claims. It is not desired to limit the invention to the exact description and operation shown and described. The spirit and scope of this invention are limited only by the spirit and scope of the following claims.